

## Optimizing Hardware Simulation and Realization of Discrete Cosine Transform Using VHDL Hardware Description Language

<sup>1</sup>Mohsen Ashourian, <sup>2</sup>Omid Sharifi-Tehrani, <sup>3</sup>Peyman moallem

<sup>1</sup>Department of Electrical Engineering, Majlesi Branch, Islamic Azad University, Majlesi, Iran.

<sup>2</sup>Young Researchers Club, Majlesi Branch, Islamic Azad University, Majlesi, Iran.

<sup>3</sup>Department of Electrical and Computer Engineering, University of Isfahan.

---

**Abstract:** Discrete cosine transform (DCT) is the fundamental part of JPEG compressor and is one of the most widely used conversion technique in digital signal processing (DSP) and image compression. Due to importance of the discrete cosine transform in JPEG standard, an algorithm is proposed that is in parallel structure thus intensify hardware implementation speed of discrete cosine transform and JPEG compression procedure. The proposed method is implemented by utilizing VHDL hardware description language in structural format and follows optimal programming tips by which, low hardware resource utilization, low latency, high throughput and high clock rate are achieved. Inputs are 8-bit long, 4 separate units are considered and CSA and CLA adders are used to realize discrete cosine transform. Working frequency for this implementation is 100 MHz and each stage delay is 10ns which is optimum in comparison with other methods. This proposed method can be easily utilized in any hardware applications such as JPEG compressor, image/signal processing and etc. by minimum change in design parameters. Also, it can be used as a hard-core in embedded systems, system on chips (SOC), system on programmable chips (SOPC) and network on chips (NOC).

**Key words:** Discrete Cosine Transform, Hardware Implementation, VHDL, Low Resource Utilization.

---

### INTRODUCTION

The discrete cosine transform (DCT) has emerged as the most popular transform for many image/video compression applications owing to its near optimal performances compared to the other transform. Its energy compaction efficiency is also greater than any other transform. The 2D-DCT, in particular, is one of the major operations in current image/video compression standards. Nowadays it is the most widely used orthogonal transform for applications including videophone, video conferencing and high definition television (HDTV). Hardware implementation of signal processing algorithms with low resource utilization, low latency and high speed has become an important issue in practical applications. But restrictions and limitations of technology in hardware realization have made some difficulties (Sharifi-Tehrani, 2011; Ghafarioun, *et al.*, 2011). The 2D-DCT is computationally intensive and as such there is a great demand for high speed, high throughput and short latency computing architectures (Sharifi-Tehrani, *et al.*, 2010). Due to the high computation requirements, the 2D-DCT processor design has been concentrated on small non-overlapping blocks (typical 8x8 or 16x16). Many 2D-DCT algorithms have been proposed to achieve computational complexity reduction and thus increase the operational speed and throughput. Several hardware design methods for the implementation of the 2D-DCT have been developed recently. (Hsia, *et al.*, 2003) proposed an algorithm and architecture to calculate the 2D inverse DCT (IDCT) directly by skipping the zero DCT coefficients, since they do not affect the transform results of the IDCT. The implementation uses relatively less hardware to achieve sufficient speed for real applications. It achieves an average pixel rate varying from 150 MHz to a maximum pixel rate of 400 MHz when using a 50 MHz clock. (Chiang, *et al.*, 2004) reported a 2D-DCT/IDCT architecture which utilizes the overlapped row-column operation, instead of the transpose memory, in order to reduce the total latency of the structure. The core processor is organized into two cascaded 1D-DCT/IDCT units and one control unit. The disadvantage of this architecture is that different structures are used for the computation of the two 1D-DCT blocks. (Fernandez, *et al.*, 2004) reported an 8\*8 2D-DCT processor using the row-column decomposition method, based on the residue number system (RNS). The processors utilize a fast cosine transform algorithm that requires a single multiplication stage for each signal path. This method has large multipliers thus resulting in more hardware and degradation of system performance. In this paper we will exert cosine conversion coefficients with 6 decimal places and a complete parallel structure is used to speed-up the implementation of discrete cosine algorithm. Thus based on the method used to calculate the multiplications and summations, coefficients are produced with a delay near to 10ns in each stage. Next section describes the general trend for the implementation of discrete cosine conversion, then the designed method for performing multiplication by using VHDL language is illustrated and in the last section the results obtained are discussed and compared.

### 2D-DCT Algorithm Optimization:

For a given 2D-spatial data sequence  $\{X_{ij}; i, j = 0, 1, 2, \dots, N-1\}$ , 2D-DCT data sequence  $\{Y_{pq}; p, q = 0, 1, 2, \dots, N-1\}$  is defined by:

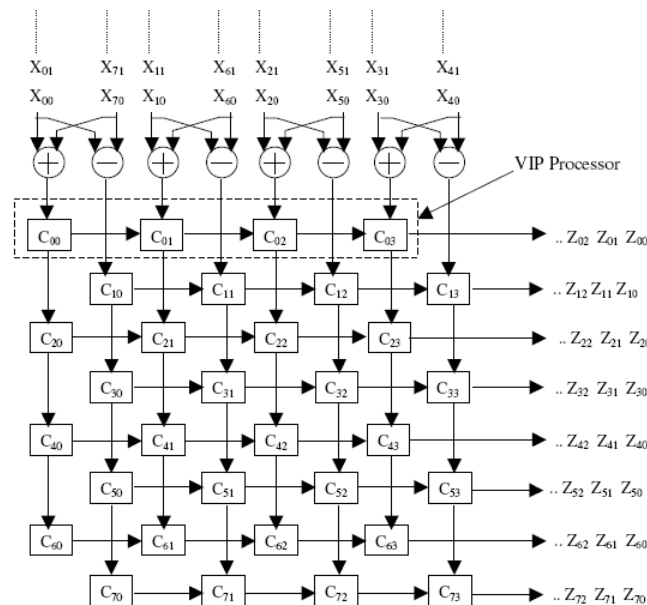
$$Y_{pq} = E_p E_q \frac{2}{N} \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} X_{ij} \cos \left[ \frac{(2i+1)p\pi}{2N} \right] \cos \left[ \frac{(2j+1)q\pi}{2N} \right] \quad (1)$$

Where

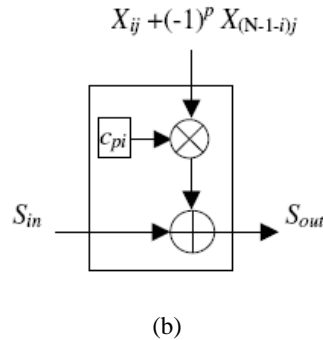
$$E_x = \begin{cases} \frac{1}{\sqrt{2}} & x=0 \\ 1 & x \neq 0 \end{cases} \quad (2)$$

We optimize the algorithm into hardware simulation by using calculation structure for 8-bit inputs shown in Figure 1. This structure includes  $N/2$  summation and subtraction units for addition and subtraction of inputs needed for equation (1). Pair inputs  $X_{ij}$  and  $X_{(N-1-i)j}$ , enter into the  $(i+1)^{th}$  summation/subtraction cell. All couples in the proposed structure entrances into summation/subtraction cells simultaneously. Considering figure 1, Structure has  $n$  units (VIP) in which, half is for the pairs that are summed together and the other half is for the pairs that are subtracted. Each VIP includes  $N/2$  multiplication/accumulator unit. Each cell saves a CPI factor in a register and helps to calculate equation (1) (Rao and Yip, 1999). CPI values are simultaneously multiplied by the corresponding data and then the results are summed together in parallel. The summation is done based on carry-bit calculation and store method within multiplication structure and is described in following sections.

In the first step, four separate units are considered as VI, VII, VIII and VII0. In the outline, there are two separate parts for each DCT units and inner vector multiplication algorithm based on  $2^N$  is utilized for multiplication operation. VII0 and VII0 parts are units in which, C weights are equal to 1 due to existence of  $\cos(0)$  and their multiplier units require individual design. In other parts, C weights may take different values. The various parts in this unit are similar in structure and the only difference is the value of C. thus, a common structure is considered for this unit. In the following, the proposed method for multiplication in DCT algorithm is described.



(a)



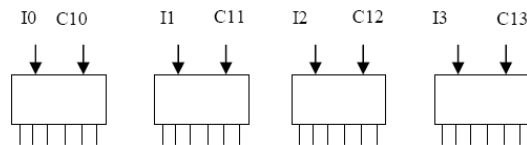
**Fig. 1:** (a) Architecture of a 1D-DCT Block for N=8. (b) Basic cell.

#### Design of VI Part:

For the first 1D-DCT, the inputs are 9-bit long and are separately multiplied by  $Cos$  weights which have 6 digits as fraction part. As an example, based on equation (1) for calculation of Z10 which is the first weight of VI unit, we have:

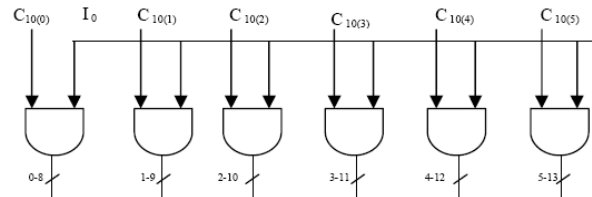
$$Z_{10} = [X_{00} - X_{70}]C_{10} + [X_{10} - X_{60}]C_{11} + [X_{20} - X_{50}]C_{12} + [X_{30} - X_{40}]C_{13} \quad (3)$$

Multiplications are done based on figure 2 and subtraction results are indicated by I0 to I3, respectively.



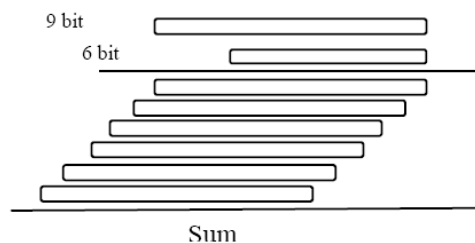
**Fig. 2:** Multiplication mode in general form of VI unit.

Each of above multiplier units has the following components in figure 3:



**Fig. 3:** Partial Plan for Multipliers of VI Unit.

According to the four multiplier blocks, there are four numbers 0 to 8 bits, 1 to 9 bits, 2 to 10 bits and so on respectively, which should be summed together to produce final result. Principal of multiplication operation is exemplified by a block diagram. For multiplication of a 9-bit long number in  $Cos$  weight which has 6 digits in fraction part, the fraction part will be denied. The corresponding multiply block diagram is depicted in figure 4. In each stage, one bit is left by placing a zero and the multiplication operation will be continued.



**Fig. 4:** Scheme of multiplying Operation in VI Unit.

Two types of adders called CLA and CSA are used to produce partial multiplication sum by inserting zero into multiplication bits to achieve unification.

#### Sum Of Partial Products:

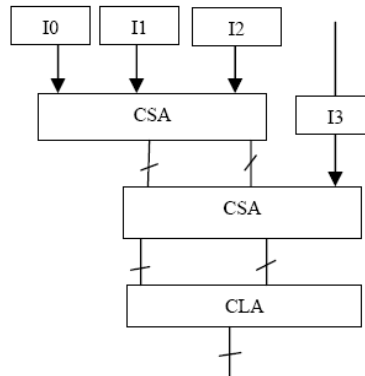
Final step is to obtain sum of partial products. To do so, we use 22 CSA units and one CLA unit to achieve each weight in first DCT. For each DCT, 7 VI units are utilized to produce different DCT weights for entering registers and multiplexers (Aggoun and Jalloh, 2005). The work flow is depicted in figure 5.

#### Design of VI0 Part:

This stage is simple due to  $Cos$  weight which is equal to 1 and is designed only with two CSA units and one CLA unit. As an example for Z00:

$$Z00 = [X00 + X70]C00 + [X10 + X60]C01 + [X20 + X50]C02 + [X30 + X41]C03 \quad (4)$$

C weights are equal to 1 in this stage. Assuming that summations of numbers are shown by I0 to I3 respectively, the block diagram of this stage is shown in figure 6.



**Fig. 6:** Block Diagram of VI0 Unit Operation.

#### Simulation Results and Discussion:

The Proposed method is simulated and synthesized using ModelSim and ISE tools, respectively. Working frequency for implementation is 100 MHz and delay time for each stage is 10 ns which is acceptable in comparison with other methods. Table 1 compares proposed method with others.

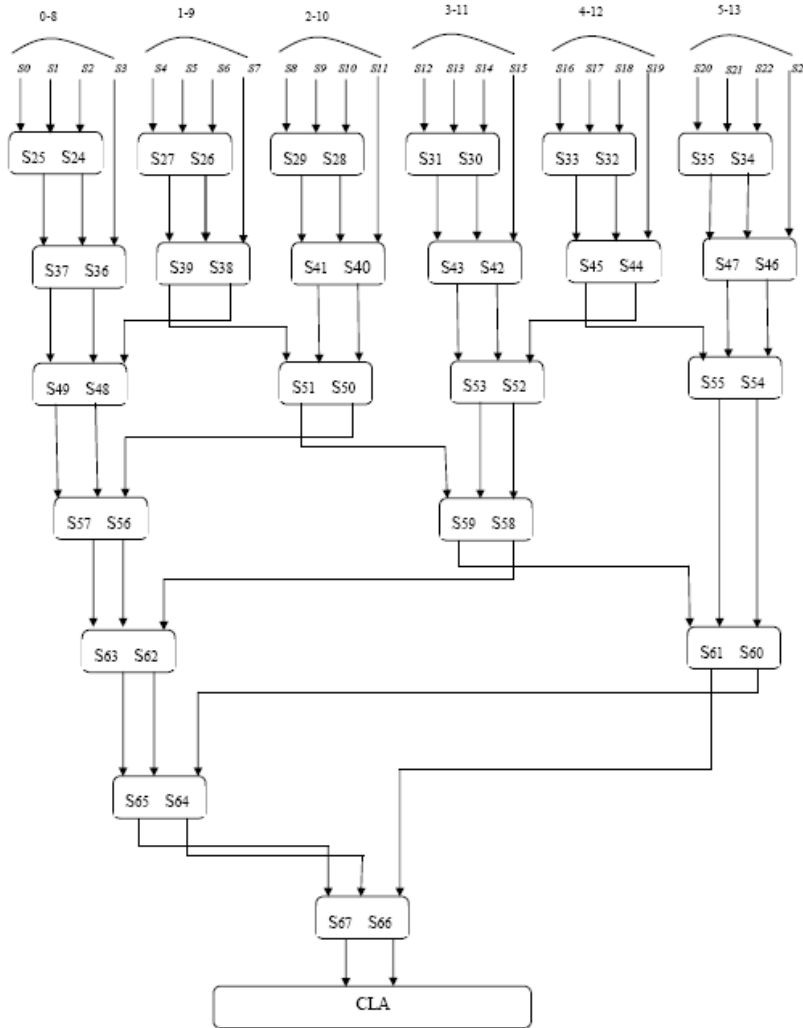
**Table 1:** Comparison between different method delays.

Method	Hsia	Chiang	Fernandez	Proposed Method
Delay (ns)	20	25	50	10
Clock (MHz)	50	40	20	100

Design procedure of VII and VII0 is completely similar to first 1D-DCT with the only different that the number of input bits is greater due to computations and thus, unit inputs are considered to be 12-bit long. The proposed method in this article is written by VHDL hardware description language in structural format and is about 4000 lines of codes. In Table 2 the optimized resource utilization using proposed method is compared with that in (Chang and Wang, 2005) for different bit lengths.

**Table 2:** Decrease in hardware resource utilization by using proposed method in comparison with that by Aggoun and Jalloh.

Bit Length	Decrease
N=4	27 %
N=8	33 %
N=16	29 %



**Fig. 5:** Block Diagram of obtaining sum of partial products in VI.

**Conclusion:**

DCT algorithm was simulated and synthesized in hardware aspect. With the definition of 1D-DCT, an approach for hardware implementation was developed. Then, a pure parallel structure for hardware realization was designed and approaches for accelerating multiplication and summation were proposed. Final results showed optimal hardware resource utilization and performance enhancement.

**REFERENCES**

- Aggoun, A. and I. Jalloh, 2003. Two Dimensional DCT/IDCT Architecture. In the IEEE Proceedings of Computer and Digital Techniques Conference, 150(1): 2-10.
- Chang, Y.T. and C.L. Wang, 2005. New Systolic Array Implementation of the 2D Discrete Cosine Transform. IEEE Trans. on Circuits and Systems for Video Tech., 5(2): 150-157.
- Chiang, J.S., Y.F. Chui and T.H. Chang, 2004. A High Throughput 2-Dimensional DCT/IDCT Architecture for Real-Time and Video System. In the IEEE Proceedings of the International Conference on Electronic Circuits and Systems, 2: 867-870.
- Fernandez, P.G., A. Garcia, J. Ramirez and A. Lioris, 2004. Fast RNS-Based 2D-DCT Computation on Field-Programmable Devices. In the IEEE Workshop on Signal Processing Systems, pp: 365-373.

Ghafarioun, E., M.Ashourian, H. Mahdavi-Nasab and O. Sharifi-Tehrani, 2011. Partial-Update Adaptive LMS Algorithms: Design, Analysis and Comparison. *Journal of International Review on Computers and Software (IReCOS)*, 6(3): 314-320.

Hsia, S.C., B.D. Liu, J.F. Yang and B.L. Bai, 2003. VLSI Implementation of Parallel Coefficient-by-Coefficient Two-Dimensional IDCT Processor. *IEEE Trans. on Circuits and Systems for Video Tech.*, 5(5): 396-406.

Rao, K.R. and P. Yip, 1999. *Discrete Cosine Transform; Algorithms, Advantages and Applications*. Academic Press Inc.

Sharifi-Tehrani, O., 2011. Novel Hardware-Efficient Design of LMS-based Adaptive FIR Filter Utilizing Finite State Machine and Block-RAM. *Journal of PRZEGLAD ELEKTROTECHNICZNY (Electrical Review)*, 87(7): 240-244.

Sharifi-Tehrani, O., M. Ashourian and P. Moallem, 2010. An FPGA-based Implementation of Fixed-Point Standard-LMS Algorithm with Low Resource Utilization and Fast Convergence. *Journal of International Review on Computers and Software (IReCOS)*, 5(4): 436-444.